

UNIVERSIDADE ESTADUAL DE SANTA CRUZ

CET058 - COMPILADORES

Leonardo Damasceno Nascimento - 201810368

Projeto 1 D – Funções Recursivas

Ilhéus/BA 2022

## Sumário

### 3. Implementação

### 4. Exemplo de códigos:

4. Fibonacci (p-code saída do código)

5. Fatorial (p-code saída do código)

5. Download do Código

5. Referências

## IMPLEMENTAÇÃO

Este relatório tem como o objetivo demonstrar um código que simula um leitor de código p-code e executa dentro da máquina/código.

Esse simulador tem como referência o código presente no artigo da wikipedia ([link](#)).

Esse código base é desenvolvido em pascal para que simule uma máquina virtual para rodar o portable code machine (P-code). A tarefa foi fazer o mesmo, porém com a linguagem C.

### Linguagem Utilizada: C

**Inputs permitidos:** Serão permitidos arquivos .txt que apresentarem itens compatíveis com a linguagem do P-code que estão pré-definidos como:

- LIT;
- OPR ;
- LOD ;
- STO ;
- CAL ;
- INT ;
- JMP;
- JPC ;

Com os exemplos do código ficará melhor a visualização do funcionamento e disposição das informações.

**Tratativa de erros:** Ao inserir um .txt que não seja compatível ele irá rodar, porém não executará da maneira desejável. Apresentando no terminal de saída:

```
start pl/0
t  b  p      f  l  a      0  1  2  3  4  5  6  7  8  9  10 11 12 13 14
=== === ===  === === ===  === === === === === === === === ===
-1  0  0  OPR  0  0      s[] :  === === === === === === === === ===
=== === ===  === === ===  === === === === === === === === ===
t  b  p      f  l  a      0  1  2  3  4  5  6  7  8  9  10 11 12 13 14
```

## Exemplos de códigos:

### 1. Fibonacci:

#### a. P-code:

1	INT	0	4
2	LIT	0	5
3	STO	0	4
4	CAL	0	6
5	LOD	0	4
6	OPR	0	0
7	INT	0	5
8	LOD	0	3
9	LIT	0	1
10	OPR	0	12
11	JP	0	14
12	LOD	0	3
13	STO	0	4
14	OPR	0	0
15	LOD	0	3
16	LIT	0	1
17	OPR	0	3
18	STO	0	5
19	CAL	0	6
20	LOD	0	5
21	STO	0	4
22	LOD	0	3
23	LIT	0	2
24	OPR	0	3
25	STO	0	5
26	CAL	0	6
27	LOD	0	5
28	LOD	0	4
29	OPR	0	2
30	STO	0	4
31	OPR	0	0

### Saída do código:

Por questão que ficou extremamente grande saída do código, será colocado apenas o final.

```

15  0 28 OPR 0 2      S[] : 0 0 0 0 0 0 0 4 5 3 4 4 28 3 1 2
14  0 29 STO 0 4      S[] : 0 0 0 0 0 0 0 4 5 3 4 4 28 3 2
13  0 30 OPR 0 0      S[] : 0 0 0 0 0 0 0 4 5 3
8   4 26 LOD 0 9      S[] : 0 0 0 0 0 0 0 4 5 3 2
9   4 27 LOD 0 4      S[] : 0 0 0 0 0 0 0 4 5 3 2 3
10  4 38 OPR 0 3      S[] : 0 0 0 0 0 0 0 4 5 3 5
9   4 29 STO 0 4      S[] : 0 0 0 0 0 0 0 4 5 3
8   4 30 OPR 0 0      S[] : 0 0 0 0
3   0 4 LOD 0 8       S[] : 0 0 0 0 5
4   0 5 OPR 0 0       S[] :
t b p # 1 a          0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
end p1/0

```

## 2. Fatorial

### a. P-code:

1	INT	0	4	14	LOD	0	3
2	LIT	0	4	15	LIT	0	1
3	STO	0	4	16	OPR	0	3
4	CAL	0	6	17	STO	0	4
5	LOD	0	4	18	CAL	0	6
6	OPR	0	0	19	LOD	0	4
7	INT	0	4	20	LOD	0	3
8	LOD	0	3	21	OPR	0	4
9	LIT	0	1	22	STO	0	3
10	OPR	0	12	23	OPR	0	0
11	JPC	0	13				
12	STO	0	3				
13	OPR	0	0				

### b. Saída do código:

```
16 12 21 STO 0 3      s[] : 0 0 0 0 0 0 4 4 4 4 18 3 8 8 18 2
15 12 22 OPR 0 0      s[] : 0 0 0 0 0 0 4 4 4 4 18 3
11 8 18 LOD 0 7       s[] : 0 0 0 0 0 0 4 4 4 4 18 3 2
12 8 19 LOD 0 3       s[] : 0 0 0 0 0 0 4 4 4 4 18 3 2 3
13 8 20 OPR 0 4       s[] : 0 0 0 0 0 0 4 4 4 4 18 3 6
12 8 21 STO 0 3       s[] : 0 0 0 0 0 0 4 4 4 4 18 6
11 8 22 OPR 0 0       s[] : 0 0 0 0 0 0 4 4
7 4 18 LOD 0 7        s[] : 0 0 0 0 0 0 4 4 6
8 4 19 LOD 0 3        s[] : 0 0 0 0 0 0 4 4 6 4
9 4 20 OPR 0 4        s[] : 0 0 0 0 0 0 4 4 24
8 4 21 STO 0 3        s[] : 0 0 0 0 0 0 4 24
7 4 22 OPR 0 0        s[] : 0 0 0 0
3 0 4 LOD 0 7         s[] : 0 0 0 0 24
4 0 5 OPR 0 0        s[] :

===
t b p f l a          0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
end 01/0
```

Link para download do código:

<https://github.com/LeoDNascimento/Compiladores/tree/main/Atividade1d>

Referências:

p-code machine. Disponível em: [https://en.wikipedia.org/wiki/P-code\\_machine](https://en.wikipedia.org/wiki/P-code_machine)